UM–SJTU Joint Institute

Introduction to Operating Systems
(VE482)

Lab 9

FUSE

Team 1

Name	ID
Xie Jinglei	517370910022
Chen Yaxin	517021911020

Date: 14 Dec. 2019

1 Concepts

1.1

A filesystem is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the filesystem.

Reference:
`https://www.tldp.org/LDP/sag/html/filesystems.html`

1.2

Linux VFS is a software level mechanics developed by C and it allows users to access various filesystems such as physical filesystems. It is an user interface for connecting kernel as well as another file system. It is a sub operating system.

A Linux VFS supports three catagories of filesystem types: disk-based filesystem, network filesystem and special filesystem. Its data structure implemented in Linux is like tree. Its root directory is just "/". In addition, there are several object structures further implemented in "linux/fs.h":

super_block: The structure stores the corresponding information of filesystem control block in disk for a disk-based filesystem. All the structure will contain a double-loop linked lists.

inode: This type of structure contains the specific file information with a unique inode number which is aimed at identifying files or directories in the filesystem. file: This structure means the file called and opened by a process, which is different from the previous two structures. The next operation will begin from the directory in the current position defined in the structure.

dentry: Kernel will assign a unique dentry structure for each part of the directory so that the file could be quickly found and efficiently edited.

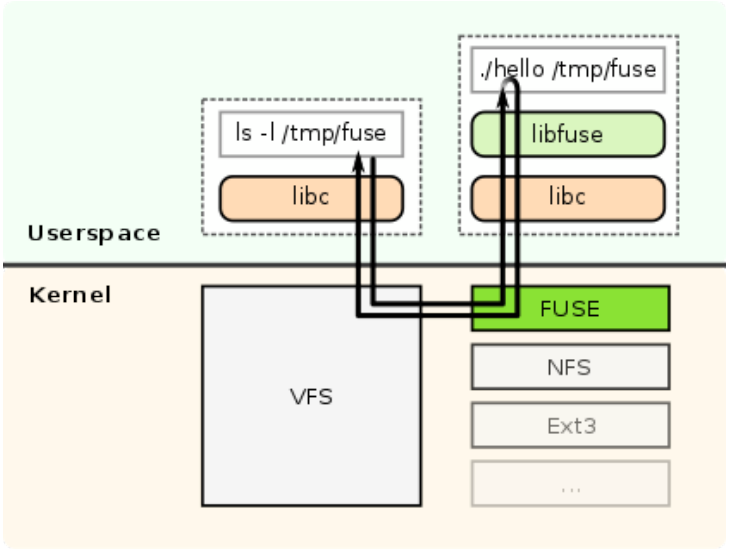
Reference: lab slides

1.3

FUSE (Filesystem in Userspace) is an interface for userspace programs to export a filesystem to the Linux kernel. The FUSE project consists of two components: the fuse kernel module (maintained in the regular kernel repositories) and the libfuse userspace library. libfuse provides the reference implementation for communicating with the FUSE kernel module. A FUSE file system is typically implemented as a standalone application that links with libfuse. libfuse provides functions to mount the file system, unmount it, read requests from the kernel, and send responses back.

By FUSE, we could directly come up with our user's own filesystem without going to kernel space, which is realized by compiling user-space code in filesystem through the functions provided by FUSE library. The FUSE system was originally part of AVFS (A Virtual Filesystem), a filesystem implementation heavily influenced by the translator concept of the GNU Hurd. Besides, we could increase the efficiency of filesystem in user space, which will significantly simplify

the cost of creating a new filesystem. Therefore, we could apply FUSE to VFS.



Above is a flow-chart diagram showing how FUSE works: Request from userspace to list files (ls -l /tmp/fuse) gets redirected by the Kernel through VFS to FUSE. FUSE then executes the registered handler program (./hello) and passes it the request (ls -l /tmp/fuse). The handler program returns a response back to FUSE which is then redirected to the userspace program that originally made the request.

Reference: lab slides
`https://libfuse.github.io/doxygen/index.html`
`https://en.wikipedia.org/wiki/Filesystem_in_Userspace`

2 Programming

Please see README.md